Segmentation of railway transport images using fuzzy logic

Balovsyak S.V.[1], Derevyanchuk O.V.[1], Derevianchuk Ya.V.[2], Tomash V.V.[1], Yarema S.V.[1]
[1]Physical, Technical and Computer Sciences Institute – Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
[2]Department "Wagon Engineering and Product Quality" – Ukrainian State University of Railway Transport, Kharkov, Ukraine

Abstract: *A prototype of a system for segmenting images of trains and wagons has been developed. Video cameras and specialized websites are used as the source of the original images. Median filtering of images and increase of their local contrast is carried out. The contours of the objects were calculated using the Sobel and Canny methods. Image segmentation is performed by the method of contour lines. As a result of the processing on the images of trains and wagons, meaningful areas (segments) were identified, for example, windows, headlights, etc. Detection of content areas of the object is performed using fuzzy membership functions. The hardware and software implementation of the computer system is made in Python using scipy and scikit-fuzzy libraries, the Google Colab cloud platform and Raspberry Pi 3B+ microcomputer.*

KEYWORDS*: DIGITAL VIDEO CAMERA, PYTHON, IMAGE SEGMENTATION, LOCAL CONTRAST, FUZZY LOGIC.*

1. Introduction

The relevance of the study is due to the fact that currently there is a need to develop automatic tools for analyzing the state of rolling stock in railway transport, namely the state of trains and wagons [1, 2]. In many cases, information about trains and wagons is obtained using video cameras in the form of digital images. To simplify image processing, content areas (segments) are highlighted on them. For example, windows, headlights, homogeneous sections of walls, license plates, wheels, etc. are distinguished as segments of locomotives. Image segmentation [3, 4] greatly simplifies their further computer processing, in particular, determining the rotation, size and area of objects, object recognition. The scope of segmentation of images of trains and wagons is quite wide. Such image processing is currently used, for example, for technical diagnostics of objects, control of their position and speed of movement. The system of analysis of the condition of vehicles on the railway has much in common with other similar transport systems, in particular, for road transport. Such systems are the basis for the development of modern intelligent transport systems, in which information about vehicles is obtained from many sensors of different types.

However, the experimental images of trains and wagons contain a certain level of noise, the images have a heterogeneous background and contrast. This leads to the fact that a significant part of the segments is uninformative. Therefore, it is proposed to perform pre-processing of images before their segmentation by filtering noise and increasing local contrast. Experimental images do not always show unambiguous correspondence of segments to certain objects. For example, the image of a locomotive window can be divided into several segments. Therefore, the paper proposes to perform object detection on images based on their segments using fuzzy functions of segment belonging to a specific object (for example, to a window).

2. Algorithm for image processing and analysis

Based on the analysis of existing methods of reducing the noise level in images, increasing local contrast, contouring and segmentation [3, 4] developed an algorithm for processing and analyzing images in the form of an activity diagram [5] (Fig. 1). Reading of initial images is carried out both from video cameras, and from graphic files. The initial color image is read as a three-dimensional array f_{RGB} (i, k, c), where $i = 0, ..., M-1$; $k = 0, ..., N-1$; M is image height in pixels, N is image width in pixels, $c = 0, ..., 2$ –color channel number (Red, Green, Blue). Color f_{RGB} images are converted to grayscale (f_0 images).

Images f_0 are programmatically processed as rectangular matrices $f_0 = (f_0 (i, k))$, where $i = 0, ..., M-1$, $k = 0, ..., N-1$.

As a result of the median filtering [3] of the image f_0, the image f with reduced noise level is calculated.

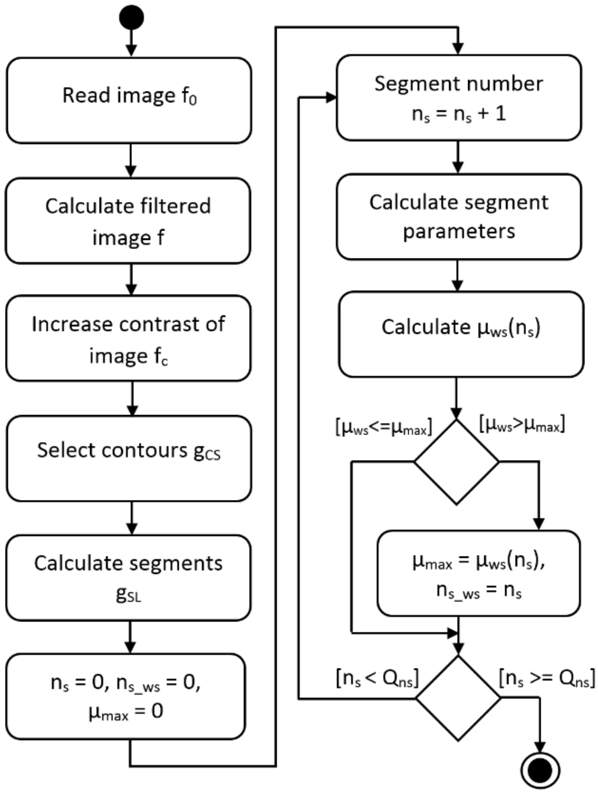


Fig. 1. *Diagram of the activity of the image segmentation system of trains and wagons*

Next, based on the filtered image f, the image f_c with high local contrast is calculated. Increasing the local contrast of images is performed by window processing of the image f using its envelope of brightness [6]. As a result of window image processing, its lower and upper envelope of brightness are first calculated, and then their smoothing is performed. Due to such smoothing distortions of the images which arise at increase in their local contrast decrease. In addition, by subtracting the lower envelope, it is possible to remove the inhomogeneous background of the image.

The sequence of increasing the local contrast is as follows. First, the dimensions of the local windows w of the image are determined, namely their height M_w and width N_w. By default, it is assumed that the windows w are square, i.e. $M_w = N_w$. When processing locally, windows w are formed for each pixel of the image f, while the pixel is placed in the center of the window at a distance of $M_{w2} = [M_w /2]$ from its upper edge and at a distance of $N_{w2} = [N_w / 2]$ from its left edge. To prevent the windows from going beyond the image, the images f are expanded before their window processing, namely the expansion in all directions by half the height M_{w2} and half the width N_{w2} of the window. The result is an expanded image of f_e. A symmetric extension is used in this work, in which stripes of width N_{w2} are added to the left and right images of f_e, and stripes of width M_{w2} are added at the top and bottom. The brightness of the bands is calculated as the brightness of the pixels of the image f, symmetrical about the boundary. Subsequently, the window processing of the extended image f_e is performed, in which the centers of the windows w are shifted by the value of M_{w2} from the edge of the image in height and by the value of N_{w2} in width. Due to this, the windows w do not go beyond the image and within each window you can correctly calculate its minimum and maximum value.

Within each window w, the minimum and maximum brightness values are calculated. The obtained values of local minima are written in the rectangular matrix of the lower envelope $f_{min} = f_{min}(i,\ k)$, where $i = 0, ..., M$-1; $k = 0, ..., N$-1. The obtained values of local maxima are written in the rectangular matrix of the upper envelope $f_{max} = f_{max}(i, k)$, where $i = 0, ..., M$-1; $k = 0, ..., N$-1.

To avoid distortion of brightness when increasing local contrast, the lower (f_{min}) and upper (f_{max}) brightness envelopes are filtered. Envelope filtration is performed by Gaussian filter with a standard deviation σ_{wG} (for example, $\sigma_{wG} = 16$) by convolving the envelopes (f_{min}, f_{max}) with the w_G core of the Gaussian filter (size $M_{wG} \times N_{wG}$ elements) according to the formulas:

$$f_{\min c}(i,k)=\sum_{m=1}^{M_{wG}}\sum_{n=1}^{N_{wG}} f_{\min}(i-m-m_c,k-n-n_c)\cdot w_G(m,n)\,, \quad (1)$$

$$f_{\max c}(i,k)=\sum_{m=1}^{M_{wG}}\sum_{n=1}^{N_{wG}} f_{\max}(i-m-m_c,k-n-n_c)\cdot w_G(m,n)\,, \quad (2)$$

where $f_{minc} = f_{minc}(i, k)$ is filtered lower envelope; $f_{maxc} = f_{maxc}(i, k)$ is filtered upper envelope; $i = 0, ..., M$-1; $k = 0, ..., N$-1; $m_c = (M_{wG2} + 1)$ is center of the filter core w_G in height; $n_c = (N_{wG2} + 1)$ is the center of the filter core w_G in width; M_{wG2}, N_{wG2} are whole parts of half the size of the core of the Gaussian filter w_G.

On the basis of envelopes (f_{minc}, f_{maxc}) and the filtered image f the image-result f_c with the increased local contrast and the removed inhomogeneous background according to the formula is calculated

$$f_c(i,k)=\frac{f(i,k)-f_{\min c}(i,k)}{f_{\max c}(i,k)-f_{\min c}(i,k)}\,, \quad (3)$$

where $i = 0, ..., M$-1; $k = 0, ..., N$-1.

Formula (3) can also be written as

$$f_c(i,k)=\left(f(i,k)-f_{\mathrm{min}c}(i,k)\right)\cdot k_c(i,k)\,, \quad (4)$$

where $k_c(i,\ k) = 1/(f_{maxc}(i,k) - f_{minc}(i,k))$ are local contrast coefficients.

In order to prevent the appearance of artifacts (for example, parasitic contours) on the restored images fc for the maximum values of the local contrast coefficients k_c set the limit k_{CMax} (for example, $k_{CMax} = 5$).

When processing color images, the pixel values of the high-contrast g_{RGB} color image are calculated by the formula

$$g_{RGB}(i,k,c)=\frac{f_c(i,k)\cdot f_{RGB}(i,k,c)}{f(i,k)}\,, \quad (5)$$

where $i = 0, ..., M$-1; $k = 0, ..., N$-1; $c = 0, ..., 2$ are color channel number; f_{RGB} is the initial color image.

In high-contrast f_c images, g_{CS} contours are calculated by Sobel and Canny methods [3]. Sobel's method provides higher speed, and Kenny's method - higher accuracy.

According to the activity diagram (Fig. 1), after reading images from the video camera, noise filtering, contrast enhancement and contour selection, segmented g_{SL} images are calculated by method of contour lines [3, 4].

The obtained segment parameters are used to assess the condition of the studied objects (trains and wagons) and to detect objects (e.g. windows) using fuzzy membership functions [7]. In the cycle by segment number n_s all segments with numbers from 1 to Q_{ns} are traversed. Segments are selected by size. Segments with a height greater than s_{iw_min} (minimum segment height) and a width greater than s_{kw_min} (minimum segment width) are selected.

Selection of objects in the image (windshield, headlights, license plates, etc.) based on their segments is performed using fuzzy sets [7]. Based on the parameters of the segments (height, width, center coordinates), the values of the fuzzy membership function μ_{ws} of the segment ns belonging to a certain object (for example, to a window) are calculated. The loop calculates the segment with the number n_{s_ws} and the maximum value of the membership function μ_{max}, because such a segment best fits the desired object in the image (for example, a window). For all segments, their normalized height s_{iwN} is calculated and the corresponding values of the function of belonging of the segment μ_{wsh} (s_{iwN}) to a certain object, taking into account its height. The values of membership functions are also calculated taking into account the normalized widths of the s_{kwN} segments and the coordinates of their centers (s_{icN}, s_{kcN}) in height and width. The value of the resulting membership function μ_{ws} of the segment with the number ns to the object, taking into account all the parameters of the segment (height, width, center coordinates in height and width) is defined as the product of the values of the corresponding membership functions:

$$\mu_{ws}(n_s)=\mu_{wsh}(s_{iwN})\cdot\mu_{wsw}(s_{kwN})\cdot\mu_{wsic}(s_{icN})\cdot\mu_{wskc}(s_{kcN})\ . \quad (6)$$

By the maximum of the membership function μ_{ws} (n_s) calculates the segment number n_{s_ws}, which most fully belongs to the specified object. Other objects in the image are detected similarly.

3. *Hardware and software implementation of image segmentation system*

2.1. *Hardware implementation of image segmentation system*

The hardware (physical) model of the train and wagons image segmentation system is constructed using the UML deployment diagram (Fig. 2).

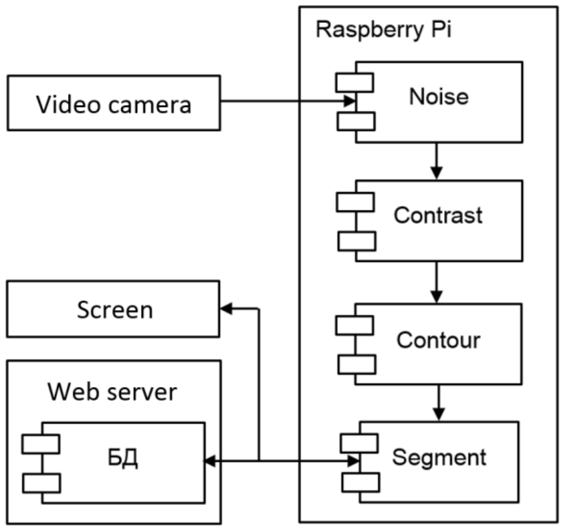


Fig. 2. *Physical model of the system of segmentation of images of trains and wagons*

The deployment diagram shows the location of system components at system nodes. According to the physical model, the initial images are read from a video camera (connected to a Raspberry Pi 3B +) and processed using a Raspberry Pi 3B + microcomputer [8]. The «Noise» software module reduces noise by filtering, the «Contrast» module increases the local contrast of the image, and the «Contour» module calculates image contours (which are then used as segment boundaries). The «Segment» module segmented images and calculated segment parameters. The «Detection» module detects objects using fuzzy membership functions. The received data is also transmitted to a remote web server for further analysis and displayed on the screen.

The hardware part of the prototype of the trains and wagons images segmentation system consists of a Raspberry Pi3B +

microcomputer [8], a digital CSI video camera or a USB video camera (Fig. 3).

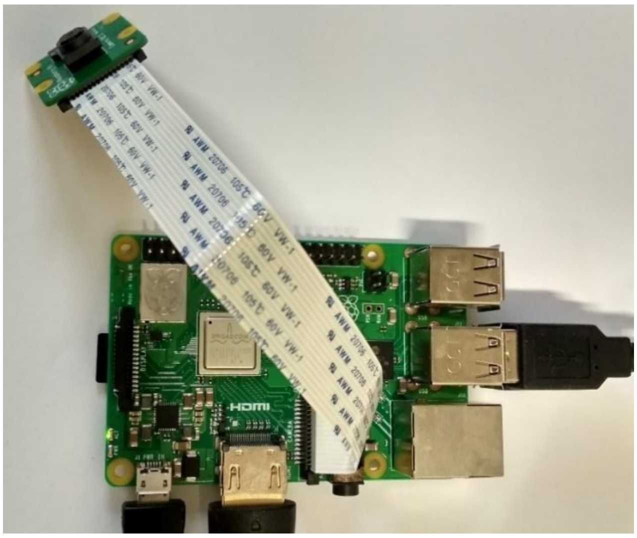


Fig. 3. *Photo of the Raspberry Pi 3 Model B + microcomputer*

2.2. Software implementation of image segmentation system

The software part of the trains and wagons images segmentation system was developed in Python first by Google Colab cloud service (in the Jupyter Notebook) [9], and then transferred to the hardware platform of the Raspberry Pi3B + microcomputer (with Raspbian operating system installed) in the development environment Thonny Python IDE [10]. The program uses libraries such as scipy, numpy, matplotlib, opencv, scikit-fuzzy.

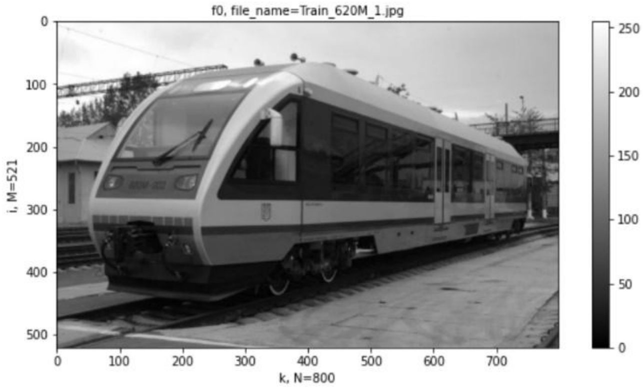
The initial (input) data for the developed program are digital images read from a video camera or from graphic files (images of trains, locomotives, wagons). The program supports reading images of basic graphic formats (.bmp, .jpg, .tiff).

The source data (results) of the program are image segments and coordinates of rectangles that correspond to the content areas of objects (for example, windows, license plates).

4. Test results of image segmentation system

Let's consider an example of segmentation of the image of a locomotive by means of the developed system. First, the initial color image is read from the graphic file (Fig. 4a) and the original image in shades of gray is calculated (Fig. 4b).

a)



b)

Fig. 4. *Initial color image f_{RGB} (a) and image f_0 in shades of gray (b) [2]*

As a result of the median filtering of the image f_0 in shades of gray (Fig. 4b), the image f is obtained with a reduced noise level. Based on the image f, the image f_c (3) with increased local contrast is calculated (Fig. 5).

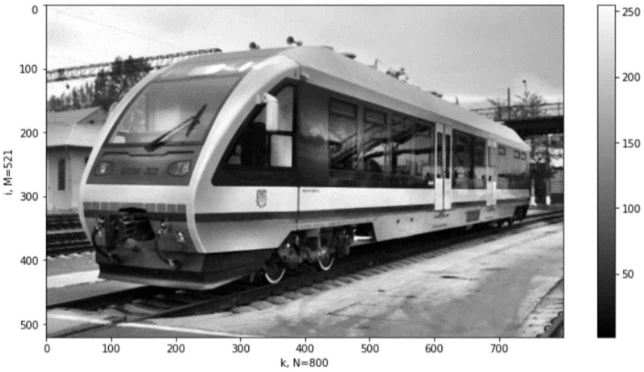


Fig. 5. *Image f_c with high local contrast, obtained on the basis of the image f*

The g_{CS} contours were calculated by the Sobel method (Fig. 6).

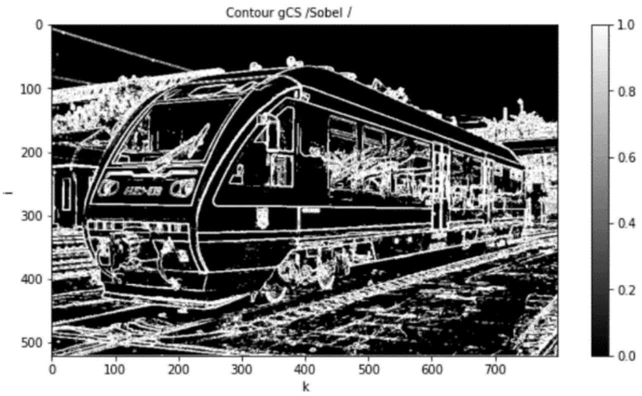


Fig. 6. Contours of g_{CS} images obtained on the basis of f_c images by Sobel method

Image segmentation is performed based on their g_{CS} contours, which highlight the boundaries of the segments. The numbers of the received segments are written in the g_{sL} array. Segments with a height greater than s_{iw_min} (minimum segment height, $s_{iw\_min} = 30$) and a width greater than s_{kw_min} (minimum segment width, $s_{kw\_min} = 30$) were selected. As a result, the most significant segments in the form of g_{sL2} images were selected (Fig. 7).

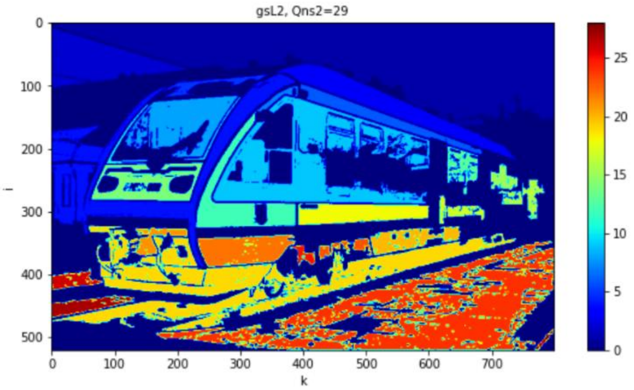


Fig. 7. *Segmented g_{sL2} image after segment selection; Q_{ns2} is number of segments*

The windshield was detected using fuzzy sets. Based on the analysis of the parameters of the segments corresponding to the windshield, the parameters of fuzzy triangular membership functions were determined (Fig. 8). For each segment with the number ns its normalized height s_{iwN} (for the image height of 1000 pixels) was determined and the corresponding value of the function of belonging of the segment μ_{wsh} (s_{iwN}) to the windshield was found taking into account its height (Fig. 8).

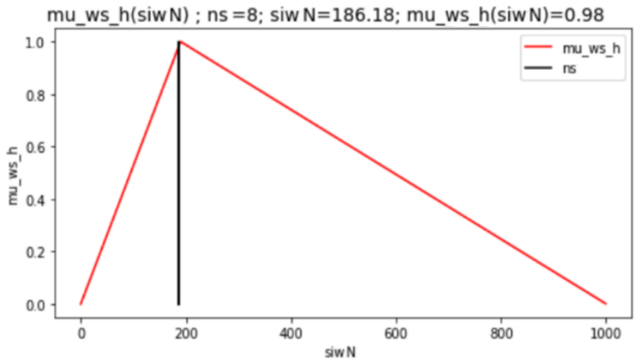


Fig. 8. *Membership functions of fuzzy sets, which describe the belonging of the segment with the number n_s to the windshield depending on the normalized height of the segment s_{iwN}*

Similarly, the values of membership functions are determined taking into account the normalized width of the segment s_{kwN} and the coordinates of its center (s_{icN}, s_{kcN}) in height and width. The value of the resulting function of belonging of the segment with the number n_s to the windshield, taking into account all the parameters of the segment is determined by formula (6). The maximum of the membership function μ_{ws} (n_s) calculates the segment number n_{s_ws}, which most fully belongs to the windshield (Fig. 9). The result is correct.

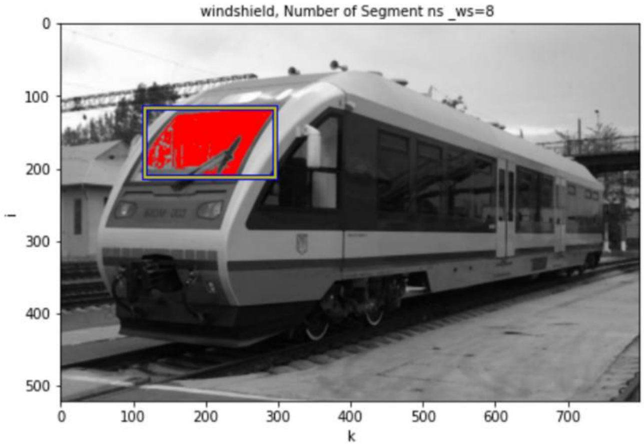


Fig. 9. *The result of windshield detection*

The results of windshield detection for other images of locomotives are also correct.

Headlights and license plates are similarly detected in the images.

5. Conclusions

1. The prototype of the system of images segmentation of trains and wagons is developed. The activity diagram and physical model of the system are developed.
2. Created a program in Python for noise filtering, increasing local contrast, contouring and image segmentation.
3. Fuzzy logic is used to detect train components in images.
4. Hardware and software implementation of the system in Python using Google Colab cloud platform and Raspberry Pi 3B + microcomputer is performed.
5. Processing of test images showed high accuracy of detection of components of trains on their images.

6. References

1. Roads Service. URL: https://www.roadsni.gov.uk.
2. Ukrzaliznytsia. URL: https://www.uz.gov.ua/press_center/photogallery/gallery-265193/.
3. Russ J.C. The Image Processing. Handbook. Abingdon-on-Thames, Taylor & Francis Group, 2011, 885 p.
4. Image Segmentation. URL: https://scikit-image.org/docs/dev/user_guide/tutorial_segmentation.html.
5. The Unified Modeling Language. URL: http://www.uml-diagrams.org.
6. Balovsyak S.V., Kravchenko H.O., Derevyanchuk O.V., Kroitor O.P., Tomash V.V. Computer system for increasing the local contrast of railway transport images. – Proc. SPIE, Fifteenth International Conference on Correlation Optics, V. 12126, 2021, P. 121261E1-7.
7. Hooda D., Raich V. Fuzzy Logic Models and Fuzzy Control. An Introduction. Alpha Science International Ltd., Oxford, U.K., 2017. 408 p.
8. Raspberry Pi Compute Module 3+. URL: https://datasheets.raspberrypi.com/cm/cm3-plus-datasheet.pdf.
9. Google Colab. URL: https://colab.research.google.com.
10. Thonny. Python IDE. URL: https://thonny.org.